



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/602,572	06/24/2003	Brian R. Tunning	MS1-1483US	1967
22801	7590	06/01/2006	EXAMINER	
LEE & HAYES PLLC 421 W RIVERSIDE AVENUE SUITE 500 SPOKANE, WA 99201			PATEL, MANGLES M	
			ART UNIT	PAPER NUMBER
			2178	

DATE MAILED: 06/01/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/602,572

Applicant(s)

TUNNING, BRIAN R.

Examiner

Manglesh M. Patel

Art Unit

2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 13 March 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-7 and 9-30 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-7 and 9-30 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 13 March 2006 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application (PTO-152)
- ☐ Other: _____

DETAILED ACTION

1. This **Final** action is responsive to the amendment filed on March 13, 2006.
2. Claims 1-7 & 9-30 are pending. Claims 1, 9, 12, 13, 15, 17, 22, 26 & 28 are independent claims.

Withdrawn Objections

3. The Objection to the specification has been withdrawn in light of the amendment.
4. The objection to the drawings has been withdrawn in light of the amendment.

Drawings

5. The newly submitted drawings filed on March 13, 2006 have been approved.

Withdrawn Rejections

6. The 35 U.S.C. 101 rejection of claims 1-16, 22-25 & 27 has been withdrawn in light of the amendment.
7. The 35 U.S.C. 102(b) rejections of claims 1-2, 9, 12, 22 and 28 with cited reference of Donohue U.S. 5,987,480 and the 35 U.S.C. 102(e) rejection of claim 26-27 with cited reference of Messler U.S. 6,970,861 and 35 U.S.C. 103(a) rejection of claims 3-8, 10-11, 13-16, 23-25 & 29-30 with cited references of Donohue U.S. 5,987,480 in

Art Unit: 2178

view of Heidingsfeld U.S. 6,823,359 have been withdrawn in light of the amendment and newly cited art.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 17-18 & 21 remain rejected under 35 U.S.C. 102(b) as being anticipated by Donohue (U.S. 5,987,480, published Nov 16, 1999).

Regarding Independent claim 17, Donohue discloses a content template system, comprising:

- A set of markup indicator tags for a web page template, wherein each tag in the set is bound to an associated element behavior (column 8, lines 40-55, wherein the dynamic tags within the web page template are bound to an associated element behavior described by the flow directive); and
- A context object to coordinate the element behaviors of the tags (column 8, lines 40-65, wherein the element behavior described by the flow directives include context object used to coordinate the behavior of the tag, see example where month Jan is used to describe the context of the flow directive).

Regarding Dependent claim 18, with dependency of claim 17,, Donohue discloses wherein a tag of the set is bound to an element behavior programmed to locate remote content referenced by an attribute of the tag, convert the remote content into a markup language used in the web page template, replace the tag with the converted content, and dynamically update the converted content in the web page template when the remote content changes (column 8, lines 10-65, wherein the dynamic tag is associated to an element behavior described by the flow directive which is used to access the remote content which includes the updating, converting and replacing of the content).

Regarding Dependent claim 21, *with dependency of claim 17*, Donohue discloses *a browser, wherein the browser renders the web page template and renders the tags bound to element behaviors* (column 2, lines 8-21, wherein a browser is used to render the templates including tags).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2178

11. Claims 19-20 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Donohue (U.S. 5,987,480, filed Jul 25, 1996) in view of Heidingsfeld (U.S. 6,823,359, filed Nov 21, 2000).

Regarding Dependent claim 19, with dependency of claim 18, Donohue teaches *wherein a tag of the set is bound to an element behavior programmed to display dynamic updates of the converted content in real-time in a web page generated by the web page template* (column 8, lines 10-40, wherein the dynamic tag is bound to an element behavior described by the flow directive to display the dynamic updates of the converted content. However Donohue fails to teach the updates of the dynamic data in real-time). Heidingsfeld teaches real-time generation of a web page based on the web page template (See Abstract). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the updating of the generated web page in real-time. The motivation for doing so would have been to allow the continuous updating of data without requiring refreshment of the web page thereby reducing user interaction. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Donohue for the benefits of allowing real-time display of dynamic data in a browser reducing user interaction for page refreshment.

Regarding Dependent claim 20, with dependency of claim 19, Donohue teaches *wherein a tag of the set is bound to an element behavior programmed to present editing controls for editing the converted content in the web page*

template and displaying the editing in real-time in the web page generated by the web page template (column 8, lines 10-40, wherein the dynamic tag is bound to an element behavior described by the flow directive to display the dynamic updates of the converted content. The editing controls are described within the template using HTML tags for character and paragraph formatting to display the content. However Donohue fails to teach the updates of the dynamic data in real-time). Heidingsfeld teaches real-time generation of a web page based on the web page template (See Abstract). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the updating of the generated web page in real-time. The motivation for doing so would have been to allow the continuous updating of data without requiring refreshment of the web page thereby reducing user interaction. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Donohue for the benefits of allowing real-time display of dynamic data in a browser reducing user interaction for page refreshment.

12. Claims 1-2, 9, 12, 22 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Donohue (U.S. 5,987,480, filed Jul 25, 1996) in view of Esposito (NPL—Element Behaviors in Internet Explorer 5.5, Dec 2000).

Regarding Independent claims 1, 12, & 28, Donohue discloses a computer-executable method, comprising: Locating remote content referenced by a tag comprising an element behavior in a web page template (See Fig 1 & column 3,

lines 45-67 & column 4, lines 1-4, wherein remote content represented by the data source is located using an embedded dynamic content tag within a web template); Converting the remote content into a markup language used in the web page template to create converted content (See Fig 1 & column 3, lines 45-67 & column 4, lines 1-4, wherein the data source representing the remote content is converted into a markup language within a web page template to create converted content using markers and directives); Replacing the tag with the converted content in response to rendering the web page template (See Abstract, wherein the document template containing the dynamic content tag is replaced by the content stored in the data source based on a delivery request which includes rendering the template); And updating the tag upon a change in the remote content or the converted content (See Abstract, It is inherent that the tag is updated when changes are made to the remote content since the content is represented by a dynamic content tag, thereby updating the tag upon changes to content). Although Donohue describes in column 8, lines 40-55, dynamic tags within the web page template are bound to an associated element behavior described by the flow directive, Esposito explicitly teaches the location of remote content referenced by a tag that includes an element behavior (pgs 4-6, section [Element versus Attached Behaviors]). Donohue teaches the use of embedded dynamic tags and flow directives for locating remote content. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content. At the time of the invention it would have been obvious to a

person or ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Esposito with Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 2, with dependency of claim 1, Donohue discloses wherein the tag has associated logic for use in at least a part of at least one of the locating, the converting, the replacing, and the updating (column 10, lines 9-20, wherein the dynamic content tags perform the locating, converting, replacing and updating using the flow directives which communicate with the data source).

Regarding Independent claim 9, Donohue discloses a computer-executable method for displaying a web page template, comprising: Executing a layout tag comprising an element behavior that indicates a style format for application to a presentation of a data content (column 8, lines 25-40, wherein the template includes the use of layout tags for describing the presentation of the data content); Executing a resource tag comprising an element behavior having logic for use in locating the data content in a remote resource, converting the data content to a markup language used in the web page template, substituting the converted content for the resource tag in the web page template, and updating

the converted content in the web page template in response to a change in the data content in the remote resource (column 8, lines 10-25, wherein the dynamic data content tag includes the use of a flow directive that represents the resource tag for the conversion/updating of the remote data content based on the changes to the content). Although Donohue describes in column 8, lines 40-55, dynamic tags within the web page template are bound to an associated element behavior described by the flow directive, Esposito explicitly teaches the location of remote content referenced by a tag that includes an element behavior (pgs 4-6, section [Element versus Attached Behaviors]). Donohue teaches the use of embedded dynamic tags and flow directives for locating remote content. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content. At the time of the invention it would have been obvious to a person or ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Esposito with Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Independent claim 22, Donohue discloses a computer-implemented content template engine, comprising: A web page processor to render a web page template having markup indicator tags into web page (See Abstract &

column 6, lines 1-35, wherein the markup indicator tags are described by the dynamic content tags and are rendered by a web page processor); A tag interpreter associated with the web page processor to obtain a logic bound to one or more of the tags(column 2, lines 5-25, wherein the tags are interpreted by the browser); A data content locator to determine a location of a remote resource having data content referenced by a tag (See Fig 1 & column 3, lines 45-67 & column 4, lines 1-4, wherein remote content represented by the data source is located using an embedded dynamic content tag within a web template); A language converter to change the data content into a language used in the web page template (See Fig 1 & column 3, lines 45-67 & column 4, lines 1-4, wherein the data source representing the remote content is converted into a markup language within a web page template to create converted content using markers and directives); A dynamic content engine to replace the tag with the converted data content and dynamically update the converted content in the web page template when the data content changes in the remote resource (See Abstract, wherein the document template containing the dynamic content tag is replaced by the content stored in the data source based on a delivery request which includes rendering the template. It is inherent that the tag is updated when changes are made to the remote content since the content is represented by a dynamic content tag, thereby updating the tag upon changes to content); Although Donohue describes in column 8, lines 40-55, dynamic tags within the web page template are bound to an associated element behavior described by

the flow directive, Esposito explicitly teaches wherein at least one of the tags comprises an element behavior (pgs 4-6, section [Element versus Attached Behaviors], wherein the tags defined by the custom HTML elements are associated to element behaviors for referencing of remote content). Donohue teaches the use of embedded dynamic tags and flow directives for locating remote content. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content. At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Esposito with Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

13. Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Klevenz (U.S. Pub 2003/0137540, filed Dec 26, 2002) in view of Lynch (U.S. 6,558,431, filed Sep 11, 1998).

Regarding Independent claim 26, Klevenz discloses in a computer system having graphical user interface including a display, a method of real-time preview of web page generation, comprising: Executing a first pane tag to implement a

first pane on the display (fig 4A, paragraphs 80-82, wherein a first pane tag is executed);

Executing a second pane tag to implement a second pane on the display (fig 4A, paragraphs 80-82, wherein a second pane tag is executed); Receiving edits of the content from the editing controls in the first pane (paragraph 84, wherein controls for editing are received in the first pane); Executing an edit tag to implement editing controls in the first pane and to display a remote content in the first pane (paragraph 84, wherein controls are displayed in the first pane);

Klevenz fails to teach the displaying of real time changes to the web page by editing of the HTML tags. Lynch discloses executing a preview tag to activate a run-time browser to perform real-time display of a web page in the second pane (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67, wherein a real-time display of a web page in a second pane is shown); Wherein the web page is generated by execution of a web page template (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67, wherein the execution of a web page template generates the web page); Wherein a resource tag in the web page template converts the remote content into custom hypertext markup language (HTML) tags (fig 11, fig 7 numeral 708, column 5, paragraphs 35-67, column 7, paragraphs 35-67, wherein a resource tag converts the remote content into HTML tags by using the HTML generator); Updating the custom HTML tags with the edits in real-time (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67, wherein updating includes displaying both the edits and content has its

modified in real time); Displaying in real time changes to the real-time execution of the web page caused by the updated HTML tags (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67, wherein the real-time execution and display of the changes with content includes a WYSIWYG editor). Klevenz teaches the generation of panes by executing a pane element. Lynch teaches a WYSIWYG editor that shows both the content and code in different panes and updates them in real-time to display them simultaneously. At the time of the invention it would have been obvious to include the updating of code with content in real-time in separate panes. The motivation for doing so would have been to implement the use of a WYSIWYG for editing of HTML visually while preserving the source document thereby allowing the preservation of format and structure of the document while allowing modeless editing. Therefore it would have been obvious to combine the teachings of Lynch with Klevenz for the benefits of real-time preview of web page generation that includes the use of a WYSIWYG editor for editing HTML and updating the content in real-time.

14. Claims 13, 14 & 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over Klevenz (U.S. Pub 2003/0137540, filed Dec 26, 2002) in view of Lynch (U.S. 6,558,431, filed Sep 11, 1998) further in view of Esposito (NPL—Element Behaviors in Internet Explorer 5.5, Dec 2000).

Regarding Independent claim 13, Klevenz teaches a computer-executable method for executing a preview tag comprising an element behavior for a web

page template, comprising: Displaying editing controls defined by the element behavior of the preview tag (paragraph 84, wherein controls for editing are received based on the execution of a preview tag); Lynch discloses reading a reference in the preview tag to at least part of the web page template to display as a currently executing web page (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67, wherein upon reading a reference such as other elements within the preview tag includes the displaying of the executing web page); Executing logic to: dynamically update the currently executing web page to display changes in content, style, and layout in part of the web page template referred to by the reference when the web page template is edited by the editing controls (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67, wherein the controls defined by Klevenz causes the dynamic update of the web page to display any changes in the content style and layout when changes are made to the HTML in real-time). Lynch fails to teach a tag that includes an element behavior. Esposito discloses at least one of the tags comprises an element behavior (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person or ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Esposito with Lynch and Klevenz for the benefits of using custom tags by

implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 14, with dependency of claim 13, Klevenz teaches the generation of panes by executing a pane element (fig 4A, paragraphs 80-82). Lynch teaches *wherein the change in content of the dynamically updated web page is displayed based on a corresponding change in content in a remote content resource accessed by the web page template* (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67). Lynch fails to teach a tag that includes an element behavior. Esposito discloses at least one of the tags comprises an element behavior (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Esposito with Lynch and Klevenz for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 27, with dependency of claim 26, Klevenz teaches the generation of panes by executing a pane element (fig 4A, paragraphs 80-82). Lynch teaches a WYSIWYG editor that shows both the content and code in

different panes and updates them in real-time to display them simultaneously (fig 11, column 5, paragraphs 35-67, column 7, paragraphs 35-67). Lynch fails to teach a tag that includes an element behavior. Esposito discloses at least one of the tags comprises an element behavior (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person or ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Esposito with Lynch and Klevenz for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

15. Claims 3-7, 10-11, 15-16, 23-25 & 29-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Donohue (U.S. 5,987,480, filed Jul 25, 1996) in view of Esposito (NPL—Element Behaviors in Internet Explorer 5.5, Dec 2000) further in view of Heidingsfeld (U.S. 6,823,359, filed Nov 21, 2000).

Regarding Dependent claims 3 & 29, Donohue teaches the use of dynamic content tags embedded within a web document template for accessing remote content (Abstract). However Donohue fails to teach the real-time updating of the generated web page. Heidingsfeld teaches *displaying updates of the tag in real-time in a web page generated by the web page template* (See Abstract, wherein dynamic data represented by the dynamic content tag is updated on a web page

in a real-time basis). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 4, with dependency of claim 3, Donohue teaches the use of dynamic content tags embedded within a web document template for accessing remote content (Abstract). However Donohue fails to teach the real-time updating of the generated web page using a preview tag. Heidingsfeld teaches *processing a preview tag in the web page template, wherein the preview tag has associated logic for use in at least a part of the displaying* (column 4, lines 14- 30, wherein the display tier represents the preview tag since it accomplishes the same task of displaying the real-time dynamic content). Heidingsfeld fails to teach element behaviors. Esposito teaches the logic associated with the preview tag comprises at least one element behavior with the

preview tag (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person or ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claims 5 & 30, Donohue teaches *presenting editing controls for editing the converted content in the web page template, wherein the editing displays one or more edits in real-time in the web page generated by the web page template* (column 3, lines 35-36, wherein the template includes editing programs that inherently contain editing controls used for designing the web page templates. However Donohue fails to teach display of a real-time web page generated by the template). Heidingsfeld teaches real-time generation of a web page based on the web page template (See Abstract). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person or ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have

been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 6, with dependency of claim 5, Donohue teaches *processing an edit tag in the web page template, wherein the edit tag has logic comprising an element behavior for use in at least a part of the editing of the converted content* (column 10, lines 9-20, wherein the web page template is designed using an editor thereby it is inherent that an edit tag is used to perform the editing of the content). Although Donohue describes in column 8, lines 40-55, dynamic tags within the web page template are bound to an associated element behavior described by the flow directive, Esposito explicitly teaches the location of remote content referenced by a tag that includes an element behavior (pgs 4-6, section [Element versus Attached Behaviors]). Donohue teaches the use of embedded dynamic tags and flow directives for locating remote content. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content. At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have

been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 7, with dependency of claim 6, Donohue teaches *displaying editing controls for the editing in one pane displayed on a computer display and simultaneously displaying the one or more edits of the converted content in real-time in the web page displayed in another pane on the computer display* (column 10, lines 9-20, wherein an editor is used to design the web page template, however Donohue fails to teach the real-time display of the edits in the template). Heidingsfeld teaches the displaying of real-time edits represented by the dynamic data in the web page template (See Abstract). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element

behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 10, with dependency of claim 9, Donohue teaches the use of dynamic content tags embedded within a web document template for accessing remote content (Abstract). However Donohue fails to teach the real-time updating of the generated web page. Heidingsfeld teaches executing a *preview tag comprising an element behavior having logic to display dynamic updates of the converted content in real-time in a web page generated by the web page template* (column 4, lines 14- 30, wherein the display tier represents the preview tag since it accomplishes the same task of displaying the real-time dynamic content, thereby providing dynamic updates of the converted content based on the web page template). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate

coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 11, with dependency of claim 10, Donohue teaches executing *an edit tag comprising an element behavior to present editing controls for editing the converted content in the web page template, wherein the editing is displayed in real-time in the web page generated by the web page template* (column 3, lines 35-36, wherein the template includes editing programs that inherently contain editing tags used for designing the web page templates. However Donohue fails to teach display of a real-time web page generated by the template). Heidingsfeld teaches real-time generation of a web page based on the web page template (See Abstract). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Independent claim 15, Donohue teaches *an edit tag comprising an element behavior for a web page template, comprising: reading a reference to at least part of a web page template to edit* (column 10, lines 9-20, wherein an editor is used to make changes to the web page template, thereby it inherently includes edit tags that reference locations within the template to apply the changes); *executing logic for: presenting editing controls for editing content in the part of the web page template referred to by the reference, wherein the editing displays results of edits in real-time in a web page generated by the part of the web page template* (Donohue teaches the use of an editor that includes editing controls for applying changes to the web page template, however Donohue fails to teach the display of the edited template in real-time). Heidingsfeld teaches real-time generation of a web page based on the web page template (See Abstract). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design

thereby reducing the complexity of the web author's task.

Regarding Dependent claim 16, with dependency of claim 15, Donohue teaches *wherein the content is dynamically updated when a corresponding content in a remote content resource accessed by the web page template changes* (See Abstract, wherein the change in content inherently includes the display since the content is represented in the template by a dynamic content tag, therefore updates to the content will directly affect the content tag).

Regarding Dependent claim 23, with dependency of claim 22, Donohue teaches the use of dynamic content tags embedded within a web document template for accessing remote content (Abstract). However Donohue fails to teach the use of a preview engine for displaying the template in real-time. Heidingsfeld teaches *a preview engine to use logic bound to a previewing tag to request a web page template and display dynamic updates of the converted content in real-time in a web page generated by the web page template, wherein the logic bound to the previewing tag comprises at least one element behavior associated with the previewing tag* (See abstract, wherein the preview engine used to display the web page template and the dynamic data in real-time is done by the display tier). Heidingsfeld teaches real-time generation of a web page based on the web page template (See Abstract). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include

custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person or ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 24, with dependency of claim 23, Donohue teaches *a module editor to use editing logic bound to an editing tag to present editing controls for changing the converted content in the web page template, wherein the results of the editing are displayed in real-time in the web page generated by the web page template, and the logic bound to the editing tag comprises at least one element behavior associated with the editing tag* (column 10, lines 9-20, wherein an editor is used to make changes to the web page template including editing controls, thereby it inherently includes edit tags that reference locations within the template to apply the changes. However Donohue fails to teach the use of an editor for displaying the content in real-time). Heidingsfeld teaches real-time generation of a web page based on the web page template (See Abstract). Heidingsfeld fails to teach element behaviors. Esposito

teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

Regarding Dependent claim 25, with dependency of claim 24, Donohue teaches the use of dynamic content tags embedded within a web document template for accessing remote content (Abstract). Including the use of an editor. However Donohue fails to teach the use of a pane controller to display the interface to edit the template simultaneously. Heidingsfeld teaches *a pane controller, to display a user interface for editing simultaneously with a display of the web page generated by the web page template* (See Abstract, although Heidingsfeld doesn't specifically describe the use of two simultaneous panes for editing and displaying, he does disclose the real-time display of the template, the editor is described by Donohue. However the display of the template is based on the changes made to the dynamic data therefore it inherently includes the

displaying of both panes to show the real-time updating of the dynamic content within the web template page). Heidingsfeld fails to teach element behaviors. Esposito teaches the use of element behaviors that include custom HTML tags for separation of script from content (pgs 4-6, section [Element versus Attached Behaviors]). At the time of the invention it would have been obvious to a person of ordinary skill in the art to include the use of element behaviors for locating remote content. The motivation for doing so would have been to provide easy separation of script from content. Therefore it would have been obvious to combine the teachings of Heidingsfeld with Esposito and Donohue for the benefits of using custom tags by implementing an element behavior to separate coding from the page design thereby reducing the complexity of the web author's task.

It is noted that any citation `[[s]]` to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. `[[See, MPEP 2123]]`

Response To Arguments

16. Applicant's arguments with respect to claims 1-16 & 22-30 have been considered but are moot in view of the new ground(s) of rejection. Although claims 17-21 still remain rejected under the Donohue reference, all other independent claims relating to element behavior have been addressed with a new reference of Esposito. In order to

Art Unit: 2178

expedite prosecution of the application claims 17-21 have not been addressed by Esposito, however the examiner strongly believes that any limitation pertaining to **element behavior** is better addressed by Esposito. It should be noted that many prior art references teaching element behaviors well before applicants filing date exist, some extra articles have been submitted with this action. These articles describe internet explorer 5.5 describing the features of element behaviors as described in the same context of the specification. Applicant argues that:

The Messler reference does not show or disclose a display which shows both the updating of HTML tags and the new related web page content as it is being edited and changed (page 28 lines 1-12).

The new cited reference of Lynch describes one of Macromedia's well-known editors that is part of Dreamweaver called WYSIWYG editor. It allows the editing and display of content simultaneously in real-time, see figure 11. If the applicant disagrees with the following comments but is willing to move the application forward via amendment to the claims please contact the examiner to schedule an interview.

Conclusion

Other Prior Art Cited

17. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Nadav Savio---NPL, Introducing DHTML Behaviors, Web monkey, pgs 1-5
- Dino Esposito---NPL, Advanced Programming With Internet Explorer 5.x, pgs 1-8

Art Unit: 2178

18. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Manglesh M. Patel whose telephone number is (571) 272-5937. The examiner can normally be reached on M,F 8:30-6:00 T,TH 8:30-3:00 Wed 8:30-7:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen S. Hong can be reached on (571)272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2178

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Manglesh M. Patel
Patent Examiner
May 23, 2006


CESAR PAULA
PRIMARY EXAMINER